

CERTIFICATE OF MAILING

Express Mail Mailing Label No. EF238909652US

Date of Deposit August 9, 2001

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents and Trademarks, Washington, DC 20231.

Mailer Lynn E. Cabiles

(print)

Mailer Lynn E. Cabiles

(signature)

FUZZY GENETIC LEARNING AUTOMATA CLASSIFIER

GOVERNMENT RIGHTS

[0001] The U.S. Government has certain rights in this invention pursuant to the clause at FAR 52.227-12.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0002] This invention relates generally to signal classifiers, and, more particularly, to a signal classifier which employs fuzzy genetic learning automata.

2. Discussion of the Related Art

[0003] Typical passive sensor systems which collect wideband signals are receiving an enormous flux of information from a continually evolving signal environment. This enormous flux of information threatens to exceed the capacities

of the human signal analysts to analyze the data. As a result, there is a pressing need for artificially intelligent systems which quickly analyze the arriving data stream, and, while so doing, effect machine learning in order to adapt to the continual signal-environmental changes. The intelligent systems should filter the signal data by identifying and separating known signals from the arriving flux of signal information. This potentially reduces the workload for the human signal analyst by allowing him/her to focus attention on signal features which are more likely to contain mission-important information.

[0004] Adaptive front-end signal classifiers are generally known in the art. Conventional front-end signal classifiers utilize many inner product-based schemes which compare the arriving signal to a set of basis functions that serve as exemplary signals. These inner product-based schemes may be based on Fourier analysis, wavelets, and/or Gabor transforms. In each of these schemes, the computational intensity is high and there is little flexibility to adapt to a changing signal environment.

[0005] Therefore, it is desirable to provide an adaptive signal classifier which learns to identify specific embedded spatiotemporal signals in an arriving signal data stream of an evolving signal environment. More specifically, it is desirable to implement a signal classifier using fuzzy automata. Fuzzy automata generally use min-max aggregation to process the incoming data, thereby obviating the need for highly intense inner product-based comparisons to exemplary signals. Since each fuzzy automata is a language-learning system, only one requirement needs to be imposed on the signals that are to be acquired: a signal needs to be

characterizable as occurring in one or more states and transitioning as a whole or partially among some or all states over time. Thus, the formal language framework of fuzzy automata is particularly well-matched to signal separation applications. In addition, the operation of each fuzzy automaton is independent of the durations of the incoming signals.

SUMMARY OF THE INVENTION

[0006] In accordance with the teachings of the present invention, a method is provided for deriving a near-optimal fuzzy automaton for a given separation problem. The method includes the steps of: (a) forming a first generation population of fuzzy automata, where the first generation population of fuzzy automata includes a plurality of fuzzy automata; (b) performing a mutation operation on each fuzzy automaton in the first generation population of fuzzy automata; (c) reproducing the first generation population of fuzzy automata using a survival of the fittest operation; and (d) applying a cross-over operator to the reproduced first generation population of fuzzy automata, thereby yielding a next-generation population of fuzzy automata. A near-optimal fuzzy automaton is identified by evaluating the performance of each fuzzy automaton in the population; otherwise the methodology is repeated until a near-optimal fuzzy automaton is derived for the given separation problem.

[0007] Additional objects, features and advantages of the present invention will become apparent from the following description and appended claims taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Figure 1 is a diagram depicting an exemplary signal classifier implemented in accordance with the present invention;

[0009] Figures 2A and 2B are flowcharts illustrating a method for deriving near-optimal fuzzy automata for a given separation problem in accordance with the present invention;

[0010] Figure 3 is a diagram depicting an exemplary crossover operation between two fuzzy automata in accordance with the present invention; and

[0011] Figure 4 is a flowchart illustrating a method for deriving a signal classifier in an evolving signal environment in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0012] Figure 1 illustrates an exemplary signal classifier 10 that is implemented using fuzzy genetic learning automata in accordance with the present invention. Each of the fuzzy genetic learning automata are dedicated to learning and separating a specific signal or a particular set of signals from an evolving signal environment. As will be apparent to one skilled in the art, two or more fuzzy automata may be used to implement the signal classifier 10. In a preferred embodiment, each of the fuzzy automata are implemented as individual software agents.

[0013] In operation, the signal classifier 10 receives a plurality of input signals 12 from a signal environment. The input signals 12 are evaluated by each of the fuzzy automata. For illustration purposes, a first automaton, $U_{1,3}$, is intended to

classify Signals 1 and 3 and distinguish them from Signal 2; whereas a second automaton, U_2 , is intended to classify Signal 2 and distinguish it from Signals 1 and 3.

Each fuzzy automaton outputs a confusion matrix, where the rows of the matrix are output vectors arranged according to the input signals. For instance, when Signal 1 is presented to the first automaton, $A_{1,3}$, the first row of the confusion matrix is ideally [1 0 0]. Similarly, when Signal 3 is presented to the first automaton, $A_{1,3}$, the third row of the confusion matrix is ideally [0 0 1]. Likewise, when Signal 2 is presented to the second automaton, A_2 , the second row of the confusion matrix is ideally [0 1 0]. Thus, the resulting output from the signal classifier 10 is an aggregate confusion matrix 14 having the appropriate rows from each of the individual automata. In this example, the first and third rows of the aggregate confusion matrix correspond to the first and third rows of the confusion matrix from the first automaton, and the second row of the aggregate confusion matrix corresponds to the second row of the confusion matrix from the second automaton. It is noteworthy that the aggregate performance of a signal classifier built from two or more fuzzy automata is typically better than the performance of each individual fuzzy automaton.

[0014] A method for deriving a near-optimal fuzzy automaton for a given separation problem is shown in Figures 2A and 2B. In accordance with the present invention, the near-optimal fuzzy automata may then serve as the basis for a signal classifier. First, a separation problem must be identified 22 from the signal environment. A separation problem, Σ , is a set of signals {Signal 0, Signal 1, ..., Signal ξ } that need to be separated from each other. Each of the signals is further defined as a vector labeled with time, such as $A_t = [a_0^t, a_1^t, \dots, a_{m-1}^t]$, $t = 0, 1, \dots, \chi$.

Near-optimal fuzzy automata can then be derived for the given separation problem. A fuzzy automaton is a well known type of mathematical object as described by George J. Klir and Bo Yuam in the text Fuzzy Sets and Fuzzy Logic, Theory and Applications, Prentice Hall, 1995. A fuzzy automaton, U , may be represented by a quintuple $U = (uS, uR, uE^0, A^*, B^*)$, where uS is an $\ell \times \ell \times m$ matrix of fuzzy state transitions for fuzzy automaton U ; uR is an $\ell \times p$ matrix of fuzzy output transitions for fuzzy automaton U ; uE^0 is an initial vector of fuzzy state memberships of length ℓ ; A^* is a space of input vectors, each vector having length m ; and B^* is a space of output vectors, each of length p .

[0015] Initially, an input set of fuzzy automata is chosen stochastically by pulling elements of S , R , and E^0 at random from the unit interval until a set of fuzzy automata with some positive separability is realized. A population of fuzzy automata, $\{{}^gU^0, {}^gU^1, \dots, {}^gU^{\pi-1}\}$, is then formed 24 by randomly selecting a subset from the input set of fuzzy automata. In this notation, a fuzzy automaton, U , belongs to a certain generation as designated by a pre-superscript, g , where g is a non-negative integer. Likewise, a post-superscript is used to designate a particular member of the population. Thus, a first generation population of fuzzy automata is represented as $\{{}^1U^0, {}^1U^1, \dots, {}^1U^{\pi-1}\}$.

[0016] A genetic algorithm is iteratively applied to the population of fuzzy automata until a near-optimal fuzzy automaton is derived for the given separation problem. In general, a genetic algorithm is a model of learning which derives its behavior from a metaphor of the processes of evolution as they occur in nature. In nature, the members of a population are derived from and represented by

chromosomes. Likewise, the members of a population of fuzzy automata must be represented in a chromosome paradigm in order to apply a genetic algorithm.

[0017] In accordance with the present invention, at least the first three elements of the quintuple U , which are uS , uR , uE^0 , are redefined 26 in chromosome form. In particular, each data value of each matrix which comprises an element of the fuzzy automaton is represented as a binary word. For example, uS is defined as follows:

$$uS = \left[\begin{array}{cccc} uS_{0,0,0} & uS_{0,1,0} & \cdots & uS_{0,\ell-1,0} \\ uS_{1,0,0} & uS_{1,1,0} & \cdots & uS_{1,\ell-1,0} \\ \vdots & \vdots & & \vdots \\ uS_{\ell-1,0,0} & uS_{\ell-1,1,0} & \cdots & uS_{\ell-1,\ell-1,0} \end{array} \right] \left[\begin{array}{cccc} uS_{0,0,1} & uS_{0,1,1} & \cdots & uS_{0,\ell-1,1} \\ uS_{1,0,1} & uS_{1,1,1} & \cdots & uS_{1,\ell-1,1} \\ \vdots & \vdots & & \vdots \\ uS_{\ell-1,0,1} & uS_{\ell-1,1,1} & \cdots & uS_{\ell-1,\ell-1,1} \end{array} \right] \\ \dots \left[\begin{array}{cccc} uS_{0,0,m-1} & uS_{0,1,m-1} & \cdots & uS_{0,\ell-1,m-1} \\ uS_{1,0,m-1} & uS_{1,1,m-1} & \cdots & uS_{1,\ell-1,m-1} \\ \vdots & \vdots & & \vdots \\ uS_{\ell-1,0,m-1} & uS_{\ell-1,1,m-1} & \cdots & uS_{\ell-1,\ell-1,m-1} \end{array} \right]$$

where each data value is $uS_{h,i,j} = (uS_{h,i,j})_0 \circ (uS_{h,i,j})_1 \circ \cdots \circ (uS_{h,i,j})_{n-1}$ such that each term $(uS_{h,i,j})_k$ in the concatenation is an element of $\{0, 1\}$ and \circ is the concatenation operator. Accordingly, each data value, $uS_{h,i,j}$, of the matrix is a chromosome, e.g., 00101...0011, of some length n . As will be apparent to one skilled in the art, uR and uE^0 can be similarly redefined in chromosome form.

[0018] A mutation operation can then be applied 28 to each of the fuzzy automata in the population. To effect mutation, a small number of data entries in S , R , and E^0 for each automaton are randomly selected. For each entry, a random location along the bit string is flipped from $0 \rightarrow 1$ or $1 \rightarrow 0$. The bit flip function, \mathfrak{I} , on the set $\{0, 1\}$ may be defined as $\mathfrak{I}(b) = (b+1)(modulo\ 2)$, where b is a member of $\{0, 1\}$. In a preferred embodiment, mutation only occurs in the semi-automaton $\{uS, uR, uE^0\}$.

[0019] For illustration purposes, consider a $2 \times 4 \times 3$ fuzzy automaton where the random entries chosen for mutation are marked with a star as follows:

$$\left\{ \left[\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array} \right] \begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array} \right], \left[\begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \right], \left[\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{array} \right] \right\}$$

Thus, there are two data entries where mutation by bit flipping occurs: $uS_{1,1,0}$ and $uR_{0,2}$. Within each data entry, a random location is then selected. For example, suppose that the third bit is flipped for $uS_{1,1,0}$ and the fifth bit is flipped for $uR_{0,2}$. The resulting mutation operation would be as follows:

$$\begin{aligned} (uS_{1,1,0})_0 \circ (uS_{1,1,0})_1 \circ (uS_{1,1,0})_2 \circ \dots \circ (uS_{1,1,0})_7 &\Rightarrow \\ (uS_{1,1,0})_0 \circ (uS_{1,1,0})_1 \circ \mathfrak{I}((uS_{1,1,0})_2) \circ \dots \circ (uS_{1,1,0})_7 \end{aligned}$$

and

$$\begin{aligned} (uR_{0,2})_0 \circ (uR_{0,2})_1 \circ (uR_{0,2})_2 \circ \dots \circ (uR_{0,2})_7 &\Rightarrow \\ (uR_{0,2})_0 \circ (uR_{0,2})_1 \circ \dots \circ \mathfrak{I}((uR_{0,2})_5) \circ (uR_{0,2})_6 \circ (uR_{0,2})_7. \end{aligned}$$

One skilled in the art will readily recognize that the above description is merely exemplary of a mutation operation. It is envisioned that other rules may be devised for effecting a mutation operation within the scope of the present invention.

[0020] Next, the first generation population of fuzzy automata is reproduced in accordance with a survival of the fittest operation. As will be more fully explained below, reproduction involves the following three steps: (1) evaluating the fitness 30 of each fuzzy automata in the population, (2) ranking the fuzzy automata 32 within the population according to the fitness of each fuzzy automata, and (3) reproducing at least one of the fuzzy automata 34 in accordance with some reproduction rule.

[0021] First, each of the fuzzy automata are evaluated for fitness via a fitness function. In a preferred embodiment of the present invention, the fitness of each fuzzy automaton is assessed using a diagonal dominance indicator. While the following description is provided with reference to a particular fitness function, it is envisioned that other known fitness functions fall within the broader aspects of the present invention.

[0022] In order to determine a diagonal dominance indicator for a fuzzy automaton, a set of input signals must be identified from the signal environment. The set of input signal are evaluated by each fuzzy automaton in the population, thereby yielding a confusion matrix for each fuzzy automaton. A diagonal dominance indicator is computed for each confusion matrix, where a figure of merit for diagonal dominance, D , is provided as follows:

$$D = (C_{\text{smallest on-diagonal}} - C_{\text{largest off-diagonal}}) / ((C_{\text{smallest on-diagonal}} + C_{\text{largest off-diagonal}})/2).$$

For illustration purposes, the diagonal dominance indicator for an exemplary confusion matrix of

$$\begin{bmatrix} 1.17 & .21 & .23 \\ .00 & 1.24 & .36 \\ .10 & .22 & 1.26 \end{bmatrix}$$

is computed as $D = (1.17 - 0.36)/((1.17+0.36)/2) = 1.06$. As will be apparent to one skilled in the art, a diagonal dominance indicator for a confusion matrix may be computed in accordance with other known figures of merit.

[0023] Second, each of the fuzzy automata in the population are ranked according to the fitness of the individual automaton, where the diagonal dominance indicator is indicative of the fitness of an automaton. More specifically, the fuzzy automata are ranked from the fuzzy automata having the largest diagonal dominance to the fuzzy automata having the smallest diagonal dominance.

[0024] Third, at least one of the fuzzy automata in the population is reproduced in accordance with a reproduction rule. In accordance with a survival of the fittest approach, the reproduction rule is fashioned such that the highest ranking fuzzy automata are reproduced and the poorest ranking automata are eliminated. For example, given a population of eight fuzzy automata, an exemplary reproduction rule may specify that the two highest ranking automata are duplicated, the four middle ranking automata are maintained, and the lowest two ranking automata are eliminated. This exemplary reproduction rule is illustrated as follows:

$$(\text{gU}^0, \text{gU}^1, \text{gU}^2, \text{gU}^3, \text{gU}^4, \text{gU}^5, \text{gU}^6, \text{gU}^7) \Rightarrow$$

$$\{ \text{gU}^0, \text{gU}^0, \text{gU}^1, \text{gU}^1, \text{gU}^2, \text{gU}^3, \text{gU}^4, \text{gU}^5 \}$$

This particular reproduction rule has the advantage of maintaining a constant

population size of eight. It is envisioned that other types of reproduction rules fall within the scope of the present invention.

[0025] Once the population of fuzzy automata is reproduced in accordance with a survival of the fittest operation, a unique crossover operator may be applied 36 to the population of fuzzy automata. In general, fuzzy automata crossover operates on two fuzzy automata, U and V , from a particular generation of automata, and produces two new fuzzy automata, W and X , belonging to a next generation of the population. More specifically, the crossover operator is specified as a transform Ξ_Ψ which maps the tuple of input automata (U, V) , to the tuple of output, next-generation automata (W, X) . In other words, the crossover operator may be expressed as

$$\Xi_\Psi((U, V)) = (W, X)$$

As will be described below, only the state transitions $.S$, the response transitions $.R$, and the initial states $.E$ are involved in the crossover operation. The spaces A^* and B^* are not involved in the crossover operation.

[0026] In the transform Ξ_Ψ , the label Ψ is an ordered triple $\Psi = (\Psi_S, \Psi_R, \Psi_{E0})$, where $\Psi_S, \Psi_R, \Psi_{E0}$ are matrices having the same dimensions as $.S, .R$, and $.E^0$. The entries in the arrays are numbers from the set $\{1, 2, \dots, n - 1\}$, such that the entries determine the alleles of the crossover. In other words, the entries determine the points along the chromosomes where crossover is to occur. These matrices are expressed as follows:

$$\Psi_S = \begin{bmatrix} k_{S;0,0,0} & k_{S;0,1,0} & \cdots & k_{S;0,\ell-1,0} \\ k_{S;1,0,0} & k_{S;1,1,0} & \cdots & k_{S;1,\ell-1,0} \\ \vdots & \vdots & & \vdots \\ k_{S;\ell-1,0,0} & k_{S;\ell-1,1,0} & \cdots & k_{S;\ell-1,\ell-1,0} \end{bmatrix} \begin{bmatrix} k_{S;0,0,1} & k_{S;0,1,1} & \cdots & k_{S;0,\ell-1,1} \\ k_{S;1,0,1} & k_{S;1,1,1} & \cdots & k_{S;1,\ell-1,1} \\ \vdots & \vdots & & \vdots \\ k_{S;\ell-1,0,1} & k_{S;\ell-1,1,1} & \cdots & k_{S;\ell-1,\ell-1,1} \end{bmatrix}$$

$$\dots \begin{bmatrix} k_{S;0,0,m-1} & k_{S;0,1,m-1} & \dots & k_{S;0,\ell-1,m-1} \\ k_{S;1,0,m-1} & k_{S;1,1,m-1} & \dots & k_{S;1,\ell-1,m-1} \\ \vdots & \vdots & & \vdots \\ k_{S;\ell-1,0,m-1} & k_{S;\ell-1,1,m-1} & \dots & k_{S;\ell-1,\ell-1,m-1} \end{bmatrix}$$

$$\Psi_R = \begin{bmatrix} k_{R;0,0} & k_{R;0,1} & \dots & k_{R;0,p-1} \\ k_{R;1,0} & k_{R;1,1} & \dots & k_{R;1,p-1} \\ \vdots & \vdots & & \vdots \\ k_{R;\ell-1,0} & k_{R;\ell-1,1} & \dots & k_{R;\ell-1,p-1} \end{bmatrix}$$

$$\Psi_{E_0} = [k_{E_0;0} \ k_{E_0;1} \ \dots \ k_{E_0;\ell-1}]$$

[0027] The operation of the transform Ξ_Ψ can be further specified in view of this definition for Ψ . In particular, each element of the fuzzy automata W and X are further expressed in terms of the elements of U and V . For instance, the first element of W , wS , is generated as follows:

$$wS_{h,i,j} = (uS_{h,i,j})_0 \circ (uS_{h,i,j})_1 \circ \dots \circ (uS_{h,i,j})_{k_{S;h,i,j}-1} \circ (vS_{h,i,j})_{k_{S;h,i,j}} \circ \dots \circ (vS_{h,i,j})_{n-1}$$

for each data value $wS_{h,i,j}$ within wS . In other words, $wS_{h,i,j}$ obtains its front portion from $uS_{h,i,j}$ and its tail portion from $vS_{h,i,j}$. It is noteworthy that $k_{S;h,i,j}$ from Ψ 's is used to identify the crossover point within the binary word that represents each data value.

[0028] The second element of W, wR , is generated similarly as follows:

$$w r_{h,i,j} = (U r_{h,i,j})_0 \circ (U r_{h,i,j})_1 \circ \dots \circ (U r_{h,i,j})_{k_{R;h,i,j}-1} \circ (V r_{h,i,j})_{k_{R;h,i,j}} \circ \dots \circ (V r_{h,i,j})_{n-1}$$

for each data value $wr_{h,i,j}$ within wR . Likewise, the third element of W, wE^0 , is generated as follows:

$$w e_{h,i,j}^0 = (U e_{h,i,j}^0)_0 \circ (U e_{h,i,j}^0)_1 \circ \dots \circ (U e_{h,i,j}^0)_{k_{E^0;h,i,j}-1} \circ (V e_{h,i,j}^0)_{k_{E^0;h,i,j}} \circ \dots \circ (V e_{h,i,j}^0)_{n-1}$$

[0029] The next generation fuzzy automaton, X, is generated in a similar manner. However, in this instance, the front portion of the chromosome comes from V and the tail portion of the chromosome comes from U. Each element of X is further expressed below. The first element of X, xS , is generated as follows:

$$x s_{h,i,j} = (V s_{h,i,j})_0 \circ (V s_{h,i,j})_1 \circ \dots \circ (V s_{h,i,j})_{k_{S;h,i,j}-1} \circ (U s_{h,i,j})_{k_{S;h,i,j}} \circ \dots \circ (U s_{h,i,j})_{n-1}$$

for each data value within xS . The second element of X, xR , is generated as follows:

$$x r_{h,i,j} = (V r_{h,i,j})_0 \circ (V r_{h,i,j})_1 \circ \dots \circ (V r_{h,i,j})_{k_{R;h,i,j}-1} \circ (U r_{h,i,j})_{k_{R;h,i,j}} \circ \dots \circ (U r_{h,i,j})_{n-1}$$

for each data value within xR . The third element of X, xE^0 , is generated as follows:

$$x e_{h,i,j}^0 = (V e_{h,i,j}^0)_0 \circ (V e_{h,i,j}^0)_1 \circ \dots \circ (V e_{h,i,j}^0)_{k_{E^0;h,i,j}-1} \circ (U e_{h,i,j}^0)_{k_{E^0;h,i,j}} \circ \dots \circ (U e_{h,i,j}^0)_{n-1}$$

for each data value of xE^0 . Thus, the crossover transform has been described in complete detail.

[0030] For illustration purposes, an exemplary crossover operation is illustrated below. Suppose U and V are as follows:

$$uS = \begin{bmatrix} 0011000000 & 0000100111 \\ 0011011111 & 0011000101 \end{bmatrix} \begin{bmatrix} 0001101010 & 0011111011 \\ 0001101001 & 0000100011 \end{bmatrix} ,$$

$$vS = \begin{bmatrix} \mathbf{0100010110} & 0010110101 \\ 0000110000 & 0001011011 \end{bmatrix} \begin{bmatrix} \mathbf{0010011011} & 0011100000 \\ 0000101011 & 0000111101 \end{bmatrix} ,$$

$uR = vR$, and $uE^0 = vE^0$. In addition, the crossover allele, Ψ , is

$$\Psi_S = \begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix} \begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix} ,$$

where Ψ_R and Ψ_{E0} are arbitrary since $uR = vR$, and $uE^0 = vE^0$. The operation of \boxtimes_Ψ on U and V to produce W and X is illustrated in Figure 3. The bits originating with V are shown in bold italic type; whereas those originating with U appear in regular type. As shown in Figure 3, W receives the front portions of the bit strings from U and the back portions of the bit strings from V, while X receives the front portions of the bit strings from V and the back portions of the bit strings from U. Although the same allele location k for each element of S was used in the above example, this is not intended as a limitation on the values of S. On the contrary, different allele locations k may be suitably used in S.

[0031] Although the present invention is not limited as such, the above-described crossover operation is applied to the first generation population of fuzzy automata by randomly selecting pairs of fuzzy automata from the population and then applying the crossover operator to each pair of fuzzy automata. For a population of eight fuzzy automata, the crossover operator is applied to four randomly selected

pairs of fuzzy automata. In this way, a next generation population of fuzzy automata have been derived for the given separation problem.

[0032] Lastly, each of the fuzzy automata in the next generation population is evaluated for performance 38 in view of the given separation problem. In a preferred embodiment, each fuzzy automaton is evaluated using the above-described fitness function. If one of the fuzzy automata exceeds some predefined performance threshold (e.g., a minimum diagonal dominance value), then algorithm is complete, thereby yielding a near-optimal fuzzy automaton 40 for the given separation problem.

On the other hand, if none of the fuzzy automata in the next-generation population exceeded the performance threshold, then the process is repeated starting with the next-generation population serving as the input to the mutation operation. The genetic algorithm of the present invention can be repeated until at least one fuzzy automaton exceeds the performance threshold associated with the given separation problem. As will be apparent to one skilled in the art, employing a genetic algorithm having the basic elements of mutation, reproduction of the fittest, and crossover guarantees convergence upon a near-optimal solution for the given separation problem. While the above description of the present invention has been provided with reference to a particular genetic algorithm, there are other known generation-level genetic operations (e.g., blended crossover, gray coding, etc.) which may be incorporated into the overall framework of the present invention.

[0033] A signal environment, $\in (\tau)$, will typically evolve over time. For an individual signal in the environment, the time variable t may be used to designate "fast time". However, the signal environment will ordinarily evolve over "slow time" τ

(which may also be referred to as epoch). Accordingly, the separation problem $\Sigma(\tau)$ will also typically evolve over time.

[0034] Referring to Figure 4, a method is provided for deriving a signal classifier in an evolving signal environment. As previously described, a single fuzzy automaton at a fixed epoch in history τ must learn to acquire a subset of signals within the environment. In addition, the fuzzy automaton must also not confuse these signals with the other signals in environment. If multiple fuzzy automata are needed to resolve the separation problem, then the separation problem $\Sigma(\tau)$ may be partitioned 42 into a set of separation problems $\{(\Sigma(\tau))_0, (\Sigma(\tau))_1, \dots, (\Sigma(\tau))_{\gamma-1}\}$, such

that $\Sigma(\tau) = \bigcup_{i=0}^{\gamma-1} (\Sigma(\tau))_i$.

[0035] In accordance with the present invention, a near-optimal fuzzy automaton may be derived 44 for each individual separation problem using the above-described genetic algorithm. Thus, each of the fuzzy automata would learn to produce positive diagonal dominance for the signals representing its part of the problem $(\Sigma(\tau))_i$, as well as not to have diagonal dominance for the rest of the signal problem $\Sigma(\tau) \setminus (\Sigma(\tau))_i$.

[0036] An aggregate confusion matrix can be formed 46 from the resulting set of near-optimal fuzzy automata. The aggregate confusion matrix is used to evaluated 48 the set of fuzzy automata in view of the overall separation problem. If the set of fuzzy automata exceeds some predefined performance threshold, then the set of fuzzy automata can be used as the basis for a signal classifier. On the other hand, if the set of fuzzy automata does not exceed the performance threshold, then

the separation problem is repartitioned, and the process is repeated. Likewise, the signal classifier is periodically re-evaluated over time as shown at 52 in order ensure that its performance meets some minimum performance threshold. If the signal classifier does not exceed the performance threshold, then the above-described methodology is reapplied to the separation problem. In this way, changes in the environment are learned as the epoch is advanced.

[0037] The foregoing discussion discloses and describes merely exemplary embodiments of the present invention. One skilled in the art will readily recognize from such discussion, and from the accompanying drawings and claims, that various changes, modifications and variations can be made therein without departing from the spirit and scope of the invention as defined in the following claims.